

FPT Approximation for Fair Minimum-Load Clustering

Sayan Bandyapadhyay ✉

Department of Informatics, University of Bergen, Norway

Fedor V. Fomin ✉ 

Department of Informatics, University of Bergen, Norway

Petr A. Golovach ✉ 

Department of Informatics, University of Bergen, Norway

Nidhi Purohit ✉

Department of Informatics, University of Bergen, Norway

Kirill Simonov ✉

Algorithms and Complexity Group, TU Wien, Austria

Abstract

In this paper, we consider the Minimum-Load k -Clustering/Facility Location (MLkC) problem where we are given a set P of n points in a metric space that we have to cluster and an integer $k > 0$ that denotes the number of clusters. Additionally, we are given a set F of cluster centers in the same metric space. The goal is to select a set $C \subseteq F$ of k centers and assign each point in P to a center in C , such that the maximum *load* over all centers is minimized. Here the load of a center is the sum of the distances between it and the points assigned to it.

Although clustering/facility location problems have rich literature, the minimum-load objective has not been studied substantially, and hence MLkC has remained a poorly understood problem. More interestingly, the problem is notoriously hard even in some special cases including the one in line metrics as shown by Ahmadian et al. [APPROX 2014, ACM Trans. Algorithms 2018]. They also show APX-hardness of the problem in the plane. On the other hand, the best-known approximation factor for MLkC is $O(k)$, even in the plane.

In this work, we study a fair version of MLkC inspired by the work of Chierichetti et al. [NeurIPS, 2017]. Here the input points are partitioned into ℓ protected groups, and only clusters that proportionally represent each group are allowed. MLkC is the special case with $\ell = 1$. For the fair version, we are able to obtain a randomized 3-approximation algorithm in $f(k, \ell) \cdot n^{O(1)}$ time. Also, our scheme leads to an improved $(1 + \epsilon)$ -approximation in the case of Euclidean norm with the same running time (depending also linearly on the dimension d). Our results imply the same approximations for MLkC with running time $f(k) \cdot n^{O(1)}$, achieving the first constant-factor FPT approximations for this problem in general and Euclidean metric spaces.

2012 ACM Subject Classification Theory of computation \rightarrow Facility location and clustering; Theory of computation \rightarrow Parameterized complexity and exact algorithms

Keywords and phrases fair clustering, load balancing, parameterized approximation

Digital Object Identifier 10.4230/LIPIcs.IPEC.2022.4

Funding The research leading to these results have been supported by the Research Council of Norway via the project BWCA (grant no. 314528), European Research Council (ERC) via grant LOPPRE, reference 819416, and Austrian Science Fund (FWF) via project P31336 (New Frontiers for Parameterized Complexity).



© Sayan Bandyapadhyay, Fedor V. Fomin, Petr A. Golovach, Nidhi Purohit, and Kirill Simonov; licensed under Creative Commons License CC-BY 4.0

17th International Symposium on Parameterized and Exact Computation (IPEC 2022).

Editors: Holger Dell and Jesper Nederlof; Article No. 4; pp. 4:1–4:14

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Clustering is the task of partitioning a set of data items into a number of groups (or clusters) such that each group contains similar items. Typically, the similarity in the clusters is modeled by a proxy objective function, which one needs to optimize. Being a fundamental computational problem in nature, clustering has a host of diverse applications in computer science and other disciplines. Consequently, the problem has been studied with several different and possibly independent objectives. Some of these became notably popular, for example, k -means, k -median, and k -center [22, 19, 5, 26]. In this paper, we consider an objective which has not been studied substantially in the literature. In particular, we consider minimum-load clustering. Here we are given a set P of points in a metric space that we have to cluster and an integer $k > 0$ that denotes the number of clusters. Additionally, we are given a set F of cluster centers in the same metric space. The goal is to select a set $C \subseteq F$ of k centers and assign each point in P to a center in C , such that the maximum *load* over all centers is minimized. Here the load of a center is the sum of the distances between it and the points assigned to it. That is, if P' is the set of points assigned to a center c , then its load is $\sum_{p \in P'} d(c, p)$, where d is the given metric. We formally refer to this problem as Minimum-Load k -Clustering (MLkC). MLkC can be used to model applications where the cost of serving the clients (or points) assigned to a facility (or center) is incurred by the facility, e.g., assigning jobs to the k best servers from a pool of servers balancing their loads.

Surprisingly, MLkC is NP-hard even if the solution set of centers C is given, via a reduction from makespan-minimization [2]. In fact, this assignment version of the problem can be shown to be NP-hard even in line metrics and for $k = 2$, via a simple reduction from the Partition problem [29]. (In Partition, given a set of integers, the goal is to partition it into two subsets such that the difference between the sums of the integers in two subsets is minimized.) Moreover, [2] proved that the problem is strongly NP-hard in line metrics (points on a line) and APX-hard in the plane. On the positive side, an $O(k)$ -approximation follows for this problem from any existing $O(1)$ -approximation for k -median [12, 5, 24, 31, 11]. This is true, as k -median minimizes the sum of the loads of the centers. Also, constant-approximations are known for MLkC in some special cases, e.g., in star metrics and line metrics. Beyond these special cases, obtaining better than $O(k)$ -approximation in polynomial time remained a notoriously hard question, even in the plane. Indeed, as explicitly pointed out by [2], MLkC is resilient to attack by the standard approximation techniques including LP rounding and local search, which has been fairly successful in obtaining good approximation algorithms for other clustering problems. Given these difficulties, we investigate whether it is possible to obtain $O(1)$ -approximation for MLkC if we allow time $f(k) \cdot n^{O(1)}$ instead of only $n^{O(1)}$, for some function $f(\cdot)$ independent of the input size n . Indeed, we study a much more general *fair* version of the problem.

Fair clustering was introduced by [14] with the goal of removing inherent biases from the regular clustering models. In this setting, we also have a sensitive or protected feature of the data points, e.g., gender or race. The goal is to obtain a clustering where the fraction of points from a traditionally underrepresented group (w.r.t. the protected feature) in every cluster is approximately equal to the fraction of points from this group in the whole dataset. For simplicity, they assumed that the protected feature can take only two values and designed fair k -center and k -median clustering algorithms in this setting. In particular, here one is given two sets of points R and B of color red and blue, respectively, and a balance parameter $t \in [0, 1]$. The objective is to find a clustering such that in every cluster O , the ratio between the number of red points and the number of blue points is at least t and at

most $1/t$, i.e., $t \leq \frac{|O \cap R|}{|O \cap B|} \leq 1/t$. Subsequently, [33] considered a general model where the protected feature can take any number of values and designed fair clustering algorithms for the k -center objective. Later, [9] and [8] independently considered a fair clustering model that generalizes the models in both [14] and [33]. In this model, we are given a partition $\{P_1, P_2, \dots, P_\ell\}$ of the input point set P and balance parameters $0 \leq \beta_i \leq \alpha_i \leq 1$ for each group $1 \leq i \leq \ell$. Then a clustering is called (α, β) -fair if the fraction of points from each group i in every cluster is at least β_i and at most α_i . In this paper, we study the (α, β) -Fair Minimum-Load k -Clustering (FMLkC) problem, where the goal is to compute an (α, β) -fair clustering that minimizes the maximum load. (For a formal definition, please see Section 2.) We note that the only clustering objectives considered in all the above mentioned works on fair clustering are k -means, k -median and k -center. To the best of our knowledge, fair clustering was not studied with the minimum-load objective before our work.

1.1 Our Results and Techniques

Considering the FMLkC problem in general and Euclidean metric spaces we obtain the following results.

► **Theorem 1 (Informal).** *There is a 3-approximation algorithm for (α, β) -Fair Minimum-Load k -Clustering in general metric spaces that runs in time $2^{\tilde{O}(k\ell^2)} n^{O(1)}$. For d -dimensional Euclidean spaces, there is a $(1 + \epsilon)$ -approximation algorithm for (α, β) -Fair Minimum-Load k -Clustering with running time $2^{\tilde{O}(k\ell^2/\epsilon^{O(1)})} n^{O(1)} d$.*

In the above theorem, the $\tilde{O}(\cdot)$ notation hides logarithmic factors. Note that all the running times are *fixed-parameter tractable* (FPT) [16] in k , ℓ and ϵ . Moreover, our results imply the same approximations for Minimum-Load k -Clustering with running times FPT in only k and ϵ , achieving the first constant-factor FPT approximations for this problem in general and Euclidean metric spaces. Note that in the Euclidean case, the running time depends only polynomially on the dimension d . Recall that no better than $O(k)$ -approximation was known before even in the plane, and this version is known to be APX-hard. Also, the reduction mentioned before from Partition eliminates the existence of an exact algorithm in time $f(k) \cdot n^{O(1)}$, unless $P \neq NP$, as MLkC in line metrics is already NP-hard when $k = 2$. In this sense, our FPT $(1 + \epsilon)$ -approximation for Euclidean spaces is tight and the best possible.

Our results are motivated by the recent FPT approximation results for constrained clustering with popular k -median and k -means objectives [15, 7]. However, these results are based on coresets construction. A coreset is a summary of the original dataset from which it is possible to retrieve a near-optimal clustering. Their main contribution is to show that it is possible to obtain coresets of size polynomial in k , $\log n$ and d . Alternatively, the input can be compressed to an almost equivalent instance of size $\text{poly}(kd \log n)$. Then one can enumerate all possible k -tuples of centers in FPT time using the coreset and output the k -tuple having the minimum clustering cost. This yields a $(1 + \epsilon)$ -approximation for Euclidean spaces and a slightly larger 3-approximation for general metric spaces due to some technical reasons. However, such a small-sized coreset is not known for our problems. Instead, we adapt approaches from [17, 23, 10] used for directly obtaining FPT approximations for constrained k -median and k -means clustering. We note that these schemes were known only in the special Euclidean case until recently [23]. All these schemes produce in FPT (in k) time a list of k -tuples of centers, such that at least one such k -tuple is a near-optimal set of centers. Using the similarity of the k -median and the minimum-load objectives, we show these approaches can be adapted for our problems as well. However, given such a k -tuple of

centers, assigning the points to the best centers or finding the optimal clustering, in our case is still NP-hard. Nevertheless, we give a Mixed-Integer Linear Programming (MILP) based $(1 + \epsilon)$ -approximation for this assignment problem that runs in time FPT in k and ℓ (in k only for MLkC). Our MILP is partly motivated by the fair k -median MILP [7]. However, our MILP and its rounding are more involved compared to that for fair k -median, especially due to the difference in the objectives. For example, if we forget about the fairness constraints, in that case the assignment algorithm for k -median is trivial: assign each point to its closest center. However, even in this case the assignment problem for MLkC is NP-hard. Also, no near-optimal assignment scheme was known in the literature (a 2-approximate assignment scheme follows from the generalized assignment problem (GAP) [34]). Thus, in this case we give a novel $(1 + \epsilon)$ -approximate assignment scheme. In this case, we do not need MILP – rounding of an LP is sufficient to obtain the desired assignment. All these schemes applied together help us achieve the desired FPT approximations.

1.2 Related Work

[18] and [4] studied the MLkC problem under the name *min-max star cover*, where $F = P$. In this setting, MLkC can be viewed as a weighted covering problem where the task is to cover the nodes of a graph by stars. Both works obtain *bicriteria* approximation for this problem where the solution returned has near-optimal load, but uses more than k centers. [2] studied several special cases of the MLkC problem (under the name Minimum-Load k -Facility Location¹). They fully resolved the status of the MLkC problem in line metrics. On the one hand, they designed a PTAS based on dynamic programming. On the other hand, they proved that this version is strongly NP-hard. They also designed a quasi-PTAS in tree metrics. Moreover, they studied a variant of the problem with client demands in star metrics.

The notion of fair clustering introduced by [14] has been studied extensively in the literature. For k -center objective, there are several polynomial-time true $O(1)$ -approximations [33, 9]. For k -median and k -means objectives, polynomial-time $O(1)$ -approximations are designed by violating the fairness constraints by an additive factor [9, 8] and true $O(d \log n)$ -approximation is known in \mathbb{R}^d for two groups [6]. On the other hand, it is possible to obtain true $O(1)$ -approximations for these two objectives if one is allowed to use $f(k, \ell) \cdot n^{O(1)}$ time [7]. Clustering problems have been studied under several other notions of fairness, e.g., see [3, 13, 27, 28, 21, 32, 1].

Organization. We define some useful notations and our problem formally in Section 2. Then we describe the assignment algorithm for the FMLkC problem in Section 3. Finally, in Section 4, we describe the full algorithms for FMLkC in details.

2 Preliminaries

We are given a set P of points in a metric space $(\mathcal{X}, d(\cdot, \cdot))$, that we have to cluster. We are also given a set F of cluster centers in the same metric space. We note that P and F are not-necessarily disjoint, and in fact, P may be equal to F . In the Euclidean version of a clustering problem, $P \subseteq \mathbb{R}^d$, $F = \mathbb{R}^d$ and $d(\cdot, \cdot)$ is the Euclidean distance.² In the metric

¹ MLkC can also be viewed as a facility location problem with zero facility opening costs where we can still open only k facilities.

² Due to the lack of better notations, we denote the dimension by d and distance function by $d(\cdot, \cdot)$.

version, we assume that F is finite. Thus, strictly speaking, the Euclidean version is not a special case of the metric version. In the metric version, we denote $|P \cup F|$ by n and in the Euclidean version, $|P|$ by n . For any integer $t \geq 1$, we denote the set $\{1, 2, \dots, t\}$ by $[t]$.

For a partition $\mathbb{O} = \{O_1, \dots, O_k\}$ of P and a set of k cluster centers $C = \{c_1, \dots, c_k\} \subset F$, we say that \mathbb{O} is a *clustering* of P with the centers c_1, \dots, c_k . We say that the *load cost* of this clustering (also, simply *cost of clustering*) is $\max_{i \in [k]} \text{cost}_{c_i}(O_i)$. Here $\text{cost}_{c_i}(O_i)$ denotes the sum-of-distances cost of the cluster O_i with the center c_i , which is $\text{cost}_{c_i}(O_i) = \sum_{x \in O_i} d(x, c_i)$. We use the following notation to denote the cost of clustering w.r.t. the set of centers C up to a permutation of the clusters,

$$\text{cost}_C(\mathbb{O}) = \min_{i_1, \dots, i_k} \max_{j \in [k]} \text{cost}_{c_j}(O_{i_j}),$$

where i_1, \dots, i_k is a permutation of $[k]$. We also denote by $\text{cost}(O_i)$ the cost of a cluster O_i with the optimal choice of a center, that is, $\text{cost}(O_i) = \min_{c \in F} \text{cost}_c(O_i)$, and by $\text{cost}(\mathbb{O})$ the optimal cost of clustering \mathbb{O} ,

$$\text{cost}(\mathbb{O}) = \min_{\substack{C \subset F \\ |C|=k}} \text{cost}_C(\mathbb{O}).$$

Alternatively, a clustering with centers in $C \subset F$ can be defined as an assignment $\varphi : P \rightarrow C$. The assignment φ then corresponds to a clustering $\{\varphi^{-1}(c)\}_{c \in C}$, and we say that the cost of the assignment φ is $\text{cost}(\varphi) = \max_{c \in C} \sum_{x \in P: \varphi(x)=c} d(x, c)$.

Now we formally define the main problem of our interest, where the goal is to find the minimum-cost clustering that satisfies the fairness constraints.

► **Definition 2.** *In the (α, β) -Fair Minimum-Load k -Clustering (FMLkC) problem, we are given a partition $\{P_1, P_2, \dots, P_\ell\}$ of P . We are also given an integer $k > 0$ and two fairness vectors $\alpha, \beta \in [0, 1]^\ell$, $\alpha = (\alpha_1, \dots, \alpha_\ell)$, $\beta = (\beta_1, \dots, \beta_\ell)$. The objective is to select a set of at most k centers $C \subset F$ and an assignment $\varphi : P \rightarrow C$ such that φ satisfies the following fairness constraints:*

$$\begin{aligned} |\{x \in P_i : \varphi(x) = c\}| &\leq \alpha_i \cdot |\{x \in P : \varphi(x) = c\}|, & \forall c \in C, \forall i \in [\ell], \\ |\{x \in P_i : \varphi(x) = c\}| &\geq \beta_i \cdot |\{x \in P : \varphi(x) = c\}|, & \forall c \in C, \forall i \in [\ell], \end{aligned}$$

and $\text{cost}(\varphi)$ is minimized among all such assignments.

Minimum-Load k -Clustering (MLkC) is a restricted case of FMLkC with $\ell = 1$, and hence there is no fairness constraints involved in this case. The (α, β) -Fair k -median problem is defined identically except there the cost is $\text{cost}(\varphi) = \sum_{c \in C} \sum_{x \in P: \varphi(x)=c} d(x, c)$.

3 Assignment Problem for FMLkC

In the (α, β) -fair assignment problem, we are additionally given a set of centers $C \subset F$ and the goal is to find an assignment $\varphi : P \rightarrow C$ such that φ satisfies the fairness constraints and $\text{cost}(\varphi) = \max_{c \in C} \sum_{x \in P: \varphi(x)=c} d(x, c)$ is minimized.

We refer to an assignment as a fair assignment if it satisfies the fairness constraints. Also, we denote the optimal cost of an (α, β) -fair assignment by OPT. In this section, for any $\epsilon > 0$, we give a $(1 + \epsilon)$ -approximation for this problem in $f(k, \ell, \epsilon) \cdot n^{O(1)}$ time for some computable function f . In particular, we solve a budgeted version of the problem where we are also given a budget B and the goal is to decide whether there is a fair assignment of cost at most B .

► **Lemma 3.** *Suppose there is an algorithm \mathcal{A} that given an instance of budgeted (α, β) -fair assignment and any $\epsilon > 0$, in $T(n, k, \ell, \epsilon)$ time, either returns a feasible assignment of cost at most $(1 + \epsilon)B$, or correctly detects that there is no feasible assignment with budget B . Then for any $\epsilon > 0$, one can obtain a $(1 + \epsilon)$ -approximation for (α, β) -fair assignment in $(k\ell)^{O(k\ell)}n^{O(1)} + O_\epsilon(\log k) \cdot T(n, k, \ell, \epsilon/3)$ time³.*

Proof. The idea is to first find a range where OPT belongs and then apply \mathcal{A} with budget within this range to find a feasible assignment. Given an instance I of (α, β) -fair assignment, first we use an algorithm (Theorem 8.2, [7]) to compute a fair assignment of the points to the centers of C that minimizes the (α, β) -fair k -median cost. This algorithm runs in time $(k\ell)^{O(k\ell)}n^{O(1)}$. Let D be the computed (α, β) -fair k -median cost returned by the algorithm. Then $D \leq k \cdot \text{OPT}$, as the optimal cost of (α, β) -fair assignment is at least $1/k$ fraction of the optimal (α, β) -fair k -median cost. Also, $\text{OPT} \leq D$, as optimal (α, β) -fair assignment cost is at most the optimal (α, β) -fair k -median cost. Hence $D/k \leq \text{OPT} \leq D$.

Let $\epsilon' = \epsilon/3$ and m be the maximum t such that $(1 + \epsilon')^t \leq D/k$. Also, let M be the minimum t such that $D \leq (1 + \epsilon')^t$. Thus $(1 + \epsilon')^m \leq \text{OPT} \leq (1 + \epsilon')^M$. We run the algorithm \mathcal{A} setting ϵ to be ϵ' for budget $B = (1 + \epsilon')^i$ where $m \leq i \leq M$, and use the binary search to find minimum i such that it returns a feasible assignment for some budget B . Let B' be the budget for which this algorithm returns a feasible assignment. Then $B' \leq (1 + \epsilon') \text{OPT}$, as any instance with budget $B \geq \text{OPT}$ is a yes-instance, and for such a B , \mathcal{A} returns a feasible assignment of cost at most $(1 + \epsilon)B$. Hence, the cost of the assignment returned by \mathcal{A} with budget B' is at most $(1 + \epsilon')^2 \text{OPT} \leq (1 + \epsilon) \text{OPT}$. As the algorithm \mathcal{A} can be used at most $O_\epsilon(\log(M - m + 1)) = O_\epsilon(\log(D/(D/k))) = O_\epsilon(\log k)$ times, the whole algorithm runs in time $(k\ell)^{O(k\ell)}n^{O(1)} + O_\epsilon(\log k) \cdot T(n, k, \ell, \epsilon/3)$. ◀

In the following, we design an LP rounding based algorithm for budgeted (α, β) -fair assignment with the properties required in the above lemma. Moreover, this algorithm runs in time $k^{O(k\ell)}\ell^{O((k\ell^2/\epsilon)\log(\ell/\epsilon))}n^{O(1)}$. Hence, we obtain the following theorem.

► **Theorem 4.** *For any $\epsilon > 0$, a $(1 + \epsilon)$ -approximation for (α, β) -fair assignment can be obtained in time $k^{O(k\ell)}\ell^{O((k\ell^2/\epsilon)\log(\ell/\epsilon))}n^{O(1)}$.*

Next, we design the algorithm for the budgeted version of (α, β) -fair assignment. Recall that in the budgeted version, we are given an instance I containing ℓ disjoint groups $\{P_i\}$ of $P = \{p_1, \dots, p_n\}$, a set of k centers $C = \{c_1, \dots, c_k\}$ and the budget B . Our algorithm first rounds each distance to a power of $(1 + \epsilon)$. Fix any center $c_i \in C$. We partition the points in P into a number of classes based on their distances from c_i . For all $p \in P$, let $\hat{d}(p, c_i) = (1 + \epsilon)^t \epsilon^2 B$ where $t = \lceil \log_{1+\epsilon}(d(p, c_i)/(\epsilon^2 B)) \rceil$. For each distinct t , let $d_t = (1 + \epsilon)^t \epsilon^2 B$. We refer to the points p with distance $\hat{d}(p, c_i) = d_t$ from c_i as the *distance class* t with respect to (w.r.t.) c_i , which is denoted by S_{it} .

► **Observation 5.** *For all $p \in P, c \in C, d(p, c) \leq \hat{d}(p, c) \leq (1 + \epsilon) \cdot d(p, c)$.*

Let I' be the new instance of budgeted (α, β) -fair assignment with the modified distance \hat{d} . Note that \hat{d} does not necessarily satisfy the triangle inequality. As \hat{d} is obtained by scaling d by at most a factor of $(1 + \epsilon)$, we have the following observation.

► **Observation 6.** *If there is a feasible assignment for I with budget B , then there is a feasible assignment for I' with budget $(1 + \epsilon)B$. Also, if there is a feasible assignment for I' with budget $(1 + \epsilon)B$, then there is a feasible assignment for I with budget $(1 + \epsilon)B$.*

³ $O_\epsilon(\cdot)$ notation hides a factor of $O(1/\epsilon)$.

By the above observation, it is sufficient to consider \hat{d} instead of d for the purpose of computing an assignment of cost at most $(1 + \epsilon)B$. Henceforth, by distance we mean \hat{d} .

Denote by φ^* a feasible assignment for I' of cost at most $(1 + \epsilon)B$ (if any). We define a point $p \in P$ to be *costly* w.r.t. a center $c \in C$ if $\hat{d}(p, c) \geq \epsilon^2 B$. Otherwise, we define the point to be *cheap* w.r.t. c . Note that the number of costly points that can be assigned to each center in φ^* is at most $(1 + \epsilon)/\epsilon^2 \leq 2/\epsilon^2$. The next observation follows from the fact that $d_t = (1 + \epsilon)^t \epsilon^2 B$.

► **Observation 7.** *For any point p and center $c \in C$ with $\hat{d}(p, c) = d_t$, $t < 0$ if p is cheap w.r.t. c , and $t \geq 0$ if p is costly w.r.t. c .*

Now, as we are shooting for an assignment of cost at most $(1 + \epsilon)B$, we can discard all the distances $\hat{d}(p, c)$ larger than $(1 + \epsilon)B$, i.e., we can assume that such a p will never be assigned to c . Without loss of generality, we assume that all the distances we have are bounded by $(1 + \epsilon)B$. Let Δ be the maximum t such that there are $p \in P$ and $c \in C$ with $\hat{d}(p, c) = d_t$ for a costly point p w.r.t. c . By our previous assumption, $\hat{d}(p, c) \leq (1 + \epsilon)B$. Thus $\Delta \leq \lceil \log_{1+\epsilon}((1 + \epsilon)B/(\epsilon^2 B)) \rceil = O((1/\epsilon) \log(1/\epsilon))$. For $1 \leq i \leq k$, $0 \leq t \leq \Delta$ and $1 \leq g \leq \ell$, let $z_{i,t,g}$ be the number of costly points $p \in P_g$ assigned to c_i in φ^* such that $\hat{d}(p, c_i) = d_t$. Note that each $z_{i,t,g} \leq 2/\epsilon^2$, as the total number of costly points assigned to a center is at most $2/\epsilon^2$. Thus the total number of distinct choices for these variables is $(2/\epsilon^2)^{k\ell\Delta} = (1/\epsilon)^{O((k\ell/\epsilon) \log(1/\epsilon))}$.

► **Observation 8.** *There are $(1/\epsilon)^{O((k\ell/\epsilon) \log(1/\epsilon))}$ distinct choices for the variables $\{z_{i,t,g} : 1 \leq i \leq k, 0 \leq t \leq \Delta, 1 \leq g \leq \ell\}$.*

As we can probe all such possible choices, we assume that we know the exact values of these variables in φ^* . Next, we describe a Mixed-Integer Linear Program (MILP) for the budgeted version of the problem, which is partly motivated by the (α, β) -fair k -median MILP [7]. For every point p_j and center c_i , we have a fractional variable x_{ij} denoting the extent up to which p_j is assigned to c_i . For every center c_i and group $g \in \{1, \dots, \ell\}$, we have an integral variable y_{gi} denoting the “weight” of the points assigned to c_i from group g . The constraints of the MILP are described as follows. Constraint 1 ensures that each point is assigned to the centers up to an extent of 1. Constraint 2 ensures that the weight assigned from each group g to each center c_i is exactly y_{gi} . Constraints 3 and 4 are fairness constraints. Constraint 5 ensures that the weight of costly points from each class t and group g assigned to each c_i is exactly the same as the guessed value $z_{i,t,g}$. Constraint 6 ensures that the total load assigned to each c_i is bounded by $(1 + \epsilon)B$. The first and the second expressions on the left hand side of this constraint are corresponding to costly and cheap points, respectively.

$$\sum_{1 \leq i \leq k} x_{ij} = 1 \quad \forall j \in [n] \quad (1)$$

$$\sum_{j \in [n]: p_j \in P_g} x_{ij} = y_{gi} \quad \forall i \in [k], \forall g \in [\ell], \quad (2)$$

$$y_{gi} \geq \beta_g \sum_{j \in [n]} x_{ij} \quad \forall i \in [k], \forall g \in [\ell] \quad (3)$$

$$y_{gi} \leq \alpha_g \sum_{j \in [n]} x_{ij} \quad \forall i \in [k], \forall g \in [\ell] \quad (4)$$

$$\sum_{p_j \in P_g \cap S_{it}} x_{ij} = z_{i,t,g} \quad \forall i \in [k], \forall t \in \{0, \dots, \Delta\}, \forall g \in [\ell] \quad (5)$$

4:8 FPT Approximation for Fair Minimum-Load Clustering

$$\sum_{g=1}^{\ell} \sum_{0 \leq t \leq \Delta} d_t z_{i,t,g} + \sum_{t < 0} d_t \sum_{p_j \in S_{it}} x_{ij} \leq (1 + \epsilon)B \quad \forall i \in [k] \quad (6)$$

$$x_{ij} \geq 0 \quad \forall j \in [n], \forall i \in [k], \quad (7)$$

$$y_{gi} \in \mathbb{Z}_{\geq 0} \quad \forall i \in [k], \forall g \in [\ell]. \quad (8)$$

Let us denote the above MILP by Fair-LP. A solution to Fair-LP is denoted by (x, y) . We note that the assignment φ^* induces a feasible solution to Fair-LP. We use the following popular and celebrated result to solve this MILP.

► **Proposition 9** ([30, 25, 20]). *An MILP with K integral variables and encoding size L , can be solved in time $K^{O(K)} L^{O(1)}$.*

As Fair-LP has $k\ell$ integral variables $\{y_{gi}\}$ and polynomial encoding size, it can be solved in $(k\ell)^{O(k\ell)} n^{O(1)}$ time. If this algorithm outputs that there is no feasible solution to Fair-LP, we conclude that I is a no-instance. Otherwise, let (x^*, y^*) denote the feasible solution returned by this algorithm. Note that although the y^* values are integral, x^* values can very well be fractional. Next, we show how to round these variables to obtain an integral solution to Fair-LP such that the load of every center is increased by an additive amount of $O(\epsilon\ell B)$ compared to its load in (x^*, y^*) .

Fix any group g . First, we show how to round the variables corresponding to the points of P_g . For this purpose, we construct a network $G_N = (V_N, E_N)$ with source S and sink T (see Figure 1). For each point $p_j \in P_g$, there is a node v_j in V_N . For each distance class t of every center c_i , there is a node w_{it} . Also, for each center c_i , there is a node u_i . For each $v_j \in V_N$, there is an arc (S, v_j) of capacity 1. For each $p_j \in P_g$ and center c_i , there is an arc (v_j, w_{it}) of capacity 1 where t is the index such that $p_j \in S_{it}$, i.e. $\hat{d}(p_j, c_i) = d_t$. For center c_i and distance class t , let $\lambda_{it}^g = \sum_{p_j \in P_g \cap S_{it}} x_{ij}^*$, i.e. the weight assigned from $P_g \cap S_{it}$ to c_i . For each node w_{it} , there is an arc (w_{it}, u_i) of capacity $\lceil \lambda_{it}^g \rceil$. Lastly, for each center c_i , there is an arc (u_i, T) of capacity y_{gi}^* .

Note that for each center c_i , the number of non-empty distance classes is at most the number of points n . Hence, the size of V_N is a polynomial in n . Also, note that the solution (x^*, y^*) projected on the points of P_g induces a feasible fractional solution for the problem of computing a flow of value $|P_g|$ in G_N .

► **Observation 10.** *The network G_N has a fractional flow of value $|P_g|$.*

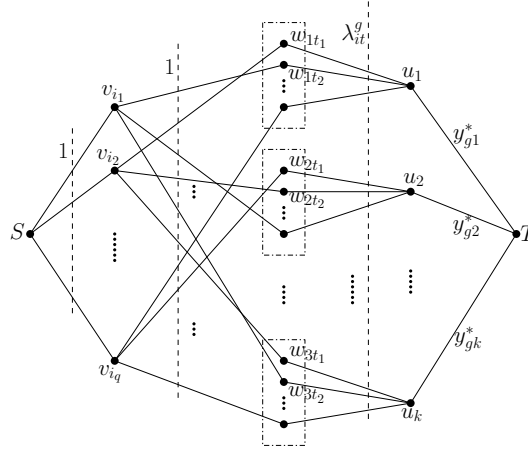
As all the capacities of the arcs are integral, by integrality of flow, there exists an integral feasible flow in G_N of value $|P_g|$. We compute such a flow f by using any polynomial time flow computation algorithm. This flow solution f naturally gives us an integral assignment φ_f of the points in P_g to the centers in C .

► **Observation 11.** *The number of points assigned to each center $c_i \in C$ via φ_f is y_{gi}^* .*

Proof. Due to the capacity constraints of the arcs $\{(u_i, T)\}$, the number of points assigned to c_i must be at most y_{gi}^* . Also, by definition, $\sum_{i=1}^k y_{gi}^* = |P_g|$. As f has value $|P_g|$, the capacity of the arcs $\{(u_i, T)\}$ must be saturated, which completes the proof. ◀

Next, we analyze the load of P_g assigned to each center via φ_f .

► **Lemma 12.** *For each center $c_i \in C$, $\sum_{p_j \in P_g: \varphi_f(p_j) = c_i} \hat{d}(p_j, c_i) \leq \sum_t d_t \lambda_{it}^g + O(\epsilon)B$.*



■ **Figure 1** Figure showing the network G_N constructed using the solution (x^*, y^*) .

Proof. Consider the arcs $\{(w_{it}, u_i)\}$. The maximum flow corresponding to these arcs is bounded by the sum of the capacities $\sum_t \lceil \lambda_{it}^g \rceil$. Note that for every $0 \leq t \leq \Delta$, $\lambda_{it}^g = z_{i,t,g}$, which is an integer. Now,

$$\sum_{t < 0} d_t = \sum_{t < 0} (1 + \epsilon)^t \epsilon^2 B < \epsilon^2 B ((1 + \epsilon)/\epsilon) = O(\epsilon)B \quad (9)$$

Hence,

$$\sum_{t < 0} d_t \lceil \lambda_{it}^g \rceil \leq \sum_{t < 0} d_t (\lambda_{it}^g + 1) \leq \sum_{t < 0} d_t \lambda_{it}^g + \sum_{t < 0} d_t \stackrel{(9)}{=} \sum_{t < 0} d_t \lambda_{it}^g + O(\epsilon)B$$

It follows that,

$$\begin{aligned} \sum_{p_j \in P_g: \varphi_f(p_j) = c_i} \hat{d}(p_j, c_i) &\leq \sum_t d_t \lceil \lambda_{it}^g \rceil = \sum_{t \geq 0} d_t \lambda_{it}^g + \sum_{t < 0} d_t \lceil \lambda_{it}^g \rceil \\ &= \sum_{t \geq 0} d_t \lambda_{it}^g + \sum_{t < 0} d_t \lambda_{it}^g + O(\epsilon)B = \sum_t d_t \lambda_{it}^g + O(\epsilon)B \quad \blacktriangleleft \end{aligned}$$

We repeat the above rounding process for all groups g . We combine the assignment functions φ_f corresponding to the ℓ disjoint groups to obtain a single assignment for the points in P . For simplicity, we also refer to this combined assignment as φ_f . By Observation 11, φ_f is feasible, as for each center c_i and each group g , the weight of the points in P_g assigned to c_i is exactly y_{gi}^* as in (x^*, y^*) . Next, we analyze the total load of each center.

► **Lemma 13.** For each center $c_i \in C$, $\sum_{p_j: \varphi_f(p_j) = c_i} \hat{d}(p_j, c_i) \leq (1 + O(\epsilon\ell))B$

Proof.

$$\begin{aligned} \sum_{p_j: \varphi_f(p_j) = c_i} \hat{d}(p_j, c_i) &= \sum_{g=1}^{\ell} \sum_{p_j \in P_g: \varphi_f(p_j) = c_i} \hat{d}(p_j, c_i) \\ &= \sum_{g=1}^{\ell} \left(\sum_t d_t \lambda_{it}^g + O(\epsilon)B \right) \quad (\text{By Lemma 12}) \\ &= \sum_{g=1}^{\ell} \left(\sum_{t \geq 0} d_t \lambda_{it}^g + \sum_{t < 0} d_t \lambda_{it}^g \right) + O(\epsilon\ell)B \end{aligned}$$

$$\begin{aligned}
&= \sum_{g=1}^{\ell} \left(\sum_{t \geq 0} d_t z_{i,t,g} + \sum_{t < 0} d_t \left(\sum_{p_j \in P_g \cap S_{it}} x_{ij}^* \right) \right) + O(\epsilon \ell) B \\
&= \left(\sum_{g=1}^{\ell} \sum_{t \geq 0} d_t z_{i,t,g} + \sum_{t < 0} d_t \sum_{p_j \in S_{it}} x_{ij}^* \right) + O(\epsilon \ell) B \\
&\leq (1 + \epsilon) B + O(\epsilon \ell) B && \text{(By Constraint 6 of Fair-LP)} \\
&= (1 + O(\epsilon \ell)) B && \blacktriangleleft
\end{aligned}$$

By scaling ϵ down by a factor of $\Omega(\ell)$, we obtain the desired approximate bound. Thus, by Observation 8, it follows that the number of distinct possible choices of the $z_{i,t,g}$ values is $(\ell/\epsilon)^{O((k\ell^2/\epsilon) \log(\ell/\epsilon))}$. For each such choice, solving the MILP and rounding takes $(k\ell)^{O(k\ell)} n^{O(1)}$ time. Thus, the algorithm for solving the budgeted version runs in time $k^{O(k\ell)} 2^{O((k\ell^2/\epsilon) \log^2(\ell/\epsilon))} n^{O(1)}$. The following lemma completes the proof of Theorem 4.

► **Lemma 14.** *The above MILP based algorithm for budgeted (α, β) -fair assignment, in time $k^{O(k\ell)} 2^{O((k\ell^2/\epsilon) \log^2(\ell/\epsilon))} n^{O(1)}$, either returns a feasible assignment of budget at most $(1 + \epsilon)B$, or correctly detects that there is no feasible assignment of budget B .*

4 Approximation Algorithms for FMLkC

In this section, we describe the FPT approximation algorithms for the FMLkC problem, both in the general metric case and in the Euclidean case. In the general metric case, we aim for a $(3 + \epsilon)$ -approximation, and in the Euclidean case for a $(1 + \epsilon)$ approximation, for a given $0 < \epsilon < 1$. Essentially, we obtain these algorithms as a combination of our assignment algorithm presented before, and known generic results for constrained clustering problems that follow the framework of [17]. Our general metric algorithm employs the result of [23], and in the Euclidean case we use the result of [10]. Both of these provide algorithms that in FPT time produce a reasonably short list of candidate sets of k centers, such that for each possible clustering of the input points one of the sets in the list provides the desired approximation, with good probability. Note that the results mentioned above are stated in fact for the k -median objective, and not the minimum-load clustering that we study in this work. However, by tweaking the error guarantees in the respective proofs we can show that these results hold in the minimum-load setting as well. Next, we present these in detail.

We start with the Euclidean case and show the following analogue of Theorem 1 in [10] for the minimum-load objective.

► **Theorem 15.** *Given a set of n points $P \subset \mathbb{R}^d$, parameters k and $0 < \epsilon < 1$, there is a randomized algorithm that in time $2^{\tilde{O}(k/\epsilon^{O(1)})} nd$ outputs a list L of $2^{\tilde{O}(k/\epsilon^{O(1)})}$ k -sized sets of centers such that for any partition $\mathbb{P}^* = \{P_1^*, \dots, P_k^*\}$ of P the following event occurs with probability at least $1/2$: there is a set C in L such that*

$$\text{cost}_C(\mathbb{P}^*) \leq (1 + \epsilon) \max_{i \in [k]} \text{cost}(P_i^*).$$

Proof. The algorithm proceeds exactly as Algorithm 5.1 in [10]. For the analysis, we observe that Bhattacharya et al. prove the following statement (follows immediately from invariant $P(i)$ in [10]): With constant probability, there is a set of centers $C = \{c_1, \dots, c_k\}$ in the output of the algorithm and the permutation i_1, \dots, i_k of the clusters in \mathbb{P}^* such that for each $j \in [k]$,

$$\text{cost}_{c_j}(P_{i_j}^*) \leq (1 + \frac{\epsilon}{2}) \cdot \text{cost}(P_{i_j}^*) + \frac{\epsilon}{2k} \cdot \sum_{i=1}^k \text{cost}(P_i^*).$$

From here it easily follows that the set of centers C achieves $(1 + \epsilon)$ -approximation of the cost of \mathbb{P}^* with respect to the minimum-load objective:

$$\text{cost}_C(\mathbb{P}^*) \leq \max_{j \in [k]} \text{cost}_{c_j}(P_{i_j}^*) \leq (1 + \frac{\epsilon}{2}) \cdot \max_{j \in [k]} \text{cost}(P_{i_j}^*) + \frac{\epsilon}{2k} \cdot \sum_{i=1}^k \text{cost}(P_i^*) \leq (1 + \epsilon) \cdot \max_{j \in [k]} \text{cost}(P_{i_j}^*),$$

$$\text{since } \sum_{i=1}^k \text{cost}(P_i^*) \leq k \max_{j \in [k]} \text{cost}(P_{i_j}^*). \quad \blacktriangleleft$$

In the general metric case, a similar result can be shown, however with the approximation factor of $(3 + \epsilon)$. Specifically, we show an analogue of Theorem 5 in [23] for the minimum-load objective. Similarly to Theorem 15, the algorithm and the analysis is identical to what is presented in [23], up to a different view on the cost upper bound.

► **Theorem 16.** *Given a set of n points P in a metric space, parameters k and $0 < \epsilon < 1$, there is a randomized algorithm that in time $(k/\epsilon)^{O(k)}n$ outputs a list L of $(k/\epsilon)^{O(k)}$ k -sized sets of centers such that for any partition $\mathbb{P}^* = \{P_1^*, \dots, P_k^*\}$ of P the following event occurs with probability at least $1/2$: there is a set C in L such that*

$$\text{cost}_C(\mathbb{P}^*) \leq (3 + \epsilon) \max_{i \in [k]} \text{cost}(P_i^*).$$

Proof. The algorithm proceeds exactly as Algorithm 1 in [23]. For the analysis, we observe that Goyal et al. prove the following statement (encapsulated by **Property-I** in [23]): With constant probability, there is a set of centers $C = \{c_1, \dots, c_k\}$ in the output of the algorithm and the permutation i_1, \dots, i_k of the clusters in \mathbb{P}^* such that for each $j \in [k]$,

$$\text{cost}_{c_j}(P_{i_j}^*) \leq (3 + \frac{\epsilon}{2}) \cdot \text{cost}(P_{i_j}^*) + \frac{\epsilon}{2k} \cdot \sum_{i=1}^k \text{cost}(P_i^*).$$

Now, analogously to the proof of Theorem 15, it follows that the set of centers C achieves $(3 + \epsilon)$ -approximation of the cost of \mathbb{P}^* with respect to the minimum-load objective:

$$\text{cost}_C(\mathbb{P}^*) \leq \max_{j \in [k]} \text{cost}_{c_j}(P_{i_j}^*) \leq (3 + \frac{\epsilon}{2}) \cdot \max_{j \in [k]} \text{cost}(P_{i_j}^*) + \frac{\epsilon}{2k} \cdot \sum_{i=1}^k \text{cost}(P_i^*) \leq (3 + \epsilon) \cdot \max_{j \in [k]} \text{cost}(P_{i_j}^*),$$

$$\text{since } \sum_{i=1}^k \text{cost}(P_i^*) \leq k \max_{j \in [k]} \text{cost}(P_{i_j}^*). \quad \blacktriangleleft$$

Now, Theorem 15 and Theorem 16 imply that for the Minimum-Load k -Clustering problem with any given set of constraints on the desired clustering, there exists a $(1 + \epsilon)$ -approximation algorithm in the Euclidean case, and a $(3 + \epsilon)$ -approximation algorithm in the general metric case. The running time is $2^{\tilde{O}(k/\epsilon^{O(1)})}(nd + T)$ for both algorithms, where T is the running time of an algorithm solving the respective assignment problem, either exact or $(1 + \epsilon)$ -approximate. In particular, combining the theorems with our approximation algorithm for (α, β) -fair assignment (Theorem 4), for the FMLkC problem we obtain a $(1 + \epsilon)$ -approximation in \mathbb{R}^d and a $(3 + \epsilon)$ -approximation in general metric in FPT time when parameterized by the number of clusters k and the number of protected groups ℓ .

► **Theorem 17.** *For any $0 < \epsilon < 1$, there exists a randomized $(1 + \epsilon)$ -approximation algorithm for (α, β) -Fair Minimum-Load k -Clustering in \mathbb{R}^d with running time $2^{\tilde{O}(k\ell^2/\epsilon^{O(1)})}n^{O(1)}d$. The same holds in general metric with the approximation factor of $(3 + \epsilon)$, where the running time becomes $2^{\tilde{O}(k\ell^2/\epsilon)}n^{O(1)}$.*

Proof. First, we deal with the Euclidean case. Fix an optimal fair min-load k -clustering $\mathbb{P}^* = \{P_1^*, \dots, P_k^*\}$ of P . Run the algorithm of Theorem 15 on P with error parameter ϵ_0 to obtain the list L of candidate sets of centers, here ϵ_0 is such that $(1 + \epsilon_0)^2 \leq (1 + \epsilon)$. In the following, assume that the event described in the statement of Theorem 15 occurs for the clustering \mathbb{P}^* , by constant number of repetitions the probability of this can be lifted arbitrarily close to one. That is, there exists a set of k centers C' in L such that

$$\text{cost}_{C'}(\mathbb{P}^*) \leq (1 + \epsilon_0) \max_{i \in [k]} \text{cost}(P_i^*). \quad (10)$$

Now, for each set of centers C in L , run the $(1 + \epsilon_0)$ -approximate assignment algorithm given by Theorem 4 on (P, C) , and choose the set of centers C'' that gives the best assignment cost among the considered sets, denote the computed assignment from P to C'' by φ . In what follows, we show that the set of centers C'' and the assignment $\varphi : P \rightarrow C''$ provide $(1 + \epsilon)$ -approximate solution to the given FMLkC instance. Denote by ψ the assignment from P to C' that the algorithm outputs,

$$\text{cost}(\varphi) \leq \text{cost}(\psi) \leq (1 + \epsilon_0) \text{cost}_{C'}(\mathbb{P}^*) \leq (1 + \epsilon_0)^2 \max_{i \in [k]} \text{cost}(P_i^*) \leq (1 + \epsilon) \max_{i \in [k]} \text{cost}(P_i^*),$$

where the first inequality is by the choice of C'' and φ , the second is by Theorem 4, and the third inequality is by (10).

Finally, we show that the running time bound holds. Invoking Theorem 15 takes time $2^{\tilde{O}(k/\epsilon_0^{O(1)})} nd$, and produces a list of $2^{\tilde{O}(k/\epsilon_0^{O(1)})}$ sets of centers. On each of them, running the algorithm of Theorem 4 takes time $k^{O(k\ell)} \ell^{O((k\ell^2/\epsilon_0) \log(\ell/\epsilon_0))} n^{O(1)} d$. Since $\epsilon_0 = O(\epsilon)$, the total running time can be bounded as

$$2^{\tilde{O}(k/\epsilon^{O(1)})} \left(nd + k^{O(k\ell)} \ell^{O((k\ell^2/\epsilon) \log(\ell/\epsilon))} n^{O(1)} d \right) = 2^{\tilde{O}(k\ell^2/\epsilon^{O(1)})} n^{O(1)} d.$$

The general metric case is identical, but to obtain the list of candidate sets of centers we use Theorem 16 instead of Theorem 15. The final cost bound changes to

$$\text{cost}(\varphi) \leq \text{cost}(\psi) \leq (1 + \epsilon_0) \text{cost}_{C'}(\mathbb{P}^*) \leq (3 + \epsilon_0) \cdot (1 + \epsilon_0) \max_{i \in [k]} \text{cost}(P_i^*) \leq (3 + \epsilon) \max_{i \in [k]} \text{cost}(P_i^*),$$

where ϵ_0 is chosen so that $(3 + \epsilon_0) \cdot (1 + \epsilon_0) \leq (3 + \epsilon)$. \blacktriangleleft

References

- 1 Mohsen Abbasi, Aditya Bhaskara, and Suresh Venkatasubramanian. Fair clustering via equitable group representations. In Madeleine Clare Elish, William Isaac, and Richard S. Zemel, editors, *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021*, pages 504–514. ACM, 2021.
- 2 Sara Ahmadian, Babak Behsaz, Zachary Friggstad, Amin Jorati, Mohammad R. Salavatipour, and Chaitanya Swamy. Approximation algorithms for minimum-load k -facility location. *ACM Trans. Algorithms*, 14(2):16:1–16:29, 2018.
- 3 Sara Ahmadian, Alessandro Epasto, Ravi Kumar, and Mohammad Mahdian. Clustering without over-representation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 267–275, 2019.
- 4 Esther M. Arkin, Refael Hassin, and Asaf Levin. Approximations for minimum and min-max vehicle routing problems. *J. Algorithms*, 59(1):1–18, 2006.

- 5 Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristics for k -median and facility location problems. *SIAM J. Comput.*, 33(3):544–562, 2004.
- 6 Arturs Backurs, Piotr Indyk, Krzysztof Onak, Baruch Schieber, Ali Vakilian, and Tal Wagner. Scalable fair clustering. In *International Conference on Machine Learning*, pages 405–413, 2019.
- 7 Sayan Bandyapadhyay, Fedor V. Fomin, and Kirill Simonov. On coresets for fair clustering in metric and euclidean spaces and their applications. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12–16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPICs*, pages 23:1–23:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICs.ICALP.2021.23.
- 8 Suman Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani. Fair algorithms for clustering. In *Advances in Neural Information Processing Systems*, pages 4954–4965, 2019.
- 9 Ioana O Bercea, Martin Groß, Samir Khuller, Aounon Kumar, Clemens Rösner, Daniel R Schmidt, and Melanie Schmidt. On the cost of essentially fair clusterings. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.
- 10 Anup Bhattacharya, Ragesh Jaiswal, and Amit Kumar. Faster Algorithms for the Constrained k -means Problem. *Theory of Computing Systems*, 62(1):93–115, 2018.
- 11 Jaroslav Byrka, Thomas W. Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k -median and positive correlation in budgeted optimization. *ACM Trans. Algorithms*, 13(2):23:1–23:31, 2017.
- 12 Moses Charikar, Sudipto Guha, Éva Tardos, and David B. Shmoys. A constant-factor approximation algorithm for the k -median problem (extended abstract). In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 1–10, 1999.
- 13 Xingyu Chen, Brandon Fain, Liang Lyu, and Kamesh Munagala. Proportionally fair clustering. In *International Conference on Machine Learning*, pages 1032–1041, 2019.
- 14 Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *Advances in Neural Information Processing Systems*, pages 5029–5037, 2017.
- 15 Vincent Cohen-Addad and Jason Li. On the fixed-parameter tractability of capacitated clustering. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9–12, 2019, Patras, Greece*, volume 132 of *LIPICs*, pages 41:1–41:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.
- 16 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.
- 17 Hu Ding and Jinhui Xu. A unified framework for clustering constrained data without locality property. *Algorithmica*, 82(4):808–852, 2020.
- 18 Guy Even, Naveen Garg, Jochen Könnemann, R. Ravi, and Amitabh Sinha. Covering graphs using trees and stars. In Sanjeev Arora, Klaus Jansen, José D. P. Rolim, and Amit Sahai, editors, *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques, 6th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2003 and 7th International Workshop on Randomization and Approximation Techniques in Computer Science, RANDOM 2003, Princeton, NJ, USA, August 24–26, 2003, Proceedings*, volume 2764 of *Lecture Notes in Computer Science*, pages 24–35. Springer, 2003.
- 19 Tomás Feder and Daniel Greene. Optimal algorithms for approximate clustering. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 434–444, 1988.

- 20 András Frank and Éva Tardos. An application of simultaneous diophantine approximation in combinatorial optimization. *Combinatorica*, 7(1):49–65, March 1987.
- 21 Mehrdad Ghadiri, Samira Samadi, and Santosh S. Vempala. Socially fair k-means clustering. In Madeleine Clare Elish, William Isaac, and Richard S. Zemel, editors, *FACCT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021*, pages 438–448. ACM, 2021.
- 22 Teofilo F Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.
- 23 Dishant Goyal, Ragesh Jaiswal, and Amit Kumar. FPT Approximation for Constrained Metric k-Median/Means. In *15th International Symposium on Parameterized and Exact Computation (IPEC 2020)*, volume 180 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 14:1–14:19, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.
- 24 Kamal Jain and Vijay V. Vazirani. Approximation algorithms for metric facility location and k -median problems using the primal-dual schema and lagrangian relaxation. *J. ACM*, 48(2):274–296, 2001.
- 25 Ravi Kannan. Minkowski’s convex body theorem and integer programming. *Mathematics of Operations Research*, 12(3):415–440, August 1987.
- 26 Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. A local search approximation algorithm for k-means clustering. *Comput. Geom.*, 28(2-3):89–112, 2004.
- 27 Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. Fair k-center clustering for data summarization. In *36th International Conference on Machine Learning, ICML 2019*, pages 5984–6003. International Machine Learning Society (IMLS), 2019.
- 28 Matthäus Kleindessner, Samira Samadi, Pranjal Awasthi, and Jamie Morgenstern. Guarantees for spectral clustering with fairness constraints. *arXiv preprint*, 2019. [arXiv:1901.08668](https://arxiv.org/abs/1901.08668).
- 29 Richard E. Korf. A complete anytime algorithm for number partitioning. *Artificial Intelligence*, 106(2):181–203, 1998.
- 30 H. W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, November 1983.
- 31 Shi Li and Ola Svensson. Approximating k-median via pseudo-approximation. *SIAM J. Comput.*, 45(2):530–547, 2016.
- 32 Yury Makarychev and Ali Vakilian. Approximation algorithms for socially fair clustering. In Mikhail Belkin and Samory Kpotufe, editors, *Conference on Learning Theory, COLT 2021, 15-19 August 2021, Boulder, Colorado, USA*, volume 134 of *Proceedings of Machine Learning Research*, pages 3246–3264. PMLR, 2021. URL: <http://proceedings.mlr.press/v134/makarychev21a.html>.
- 33 Clemens Rösner and Melanie Schmidt. Privacy preserving clustering with constraints. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.
- 34 David B. Shmoys and Éva Tardos. An approximation algorithm for the generalized assignment problem. *Math. Program.*, 62:461–474, 1993. doi:10.1007/BF01585178.